

Ruby - Bug #15647

Disallow "in" as a keyword argument name

03/08/2019 11:06 PM - localhostdotdev (localhost .dev)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 2.5.1p57 (2018-03-29 revision 63029) [x86_64-darwin17]	Backport: 2.4: UNKNOWN, 2.5: UNKNOWN, 2.6: UNKNOWN
Description <p>"in" is a reserved keyword and thus can't be accessed directly anyway.</p> <p>e.g. the following is a syntax error:</p> <pre>def m(in:); p in; end; m(in: 1)</pre> <p>And as usual, doing <code>in = 1</code> doesn't work.</p> <p>"in" could still be passed as a keyword argument if using some kind of catch-all, e.g. <code>m(args)</code>, <code>m(*args)</code>, etc.</p> <p>Source: https://twitter.com/drbrain/status/1104152696339087361</p>		

History

#1 - 03/08/2019 11:14 PM - drbrain (Eric Hodel)

This would break the Kernel#spawn family's backward compatibility make wrapping methods in this family harder than it already is, as you can work around this with Binding#local_variable_get.

I think this should be rejected.

#2 - 03/08/2019 11:36 PM - nobu (Nobuyoshi Nakada)

- Description updated

- Status changed from Open to Rejected

It is known that reserved words can be keyword arguments.
And Binding#local_variable_get has been added for that purpose, if argument for instance.